

The Love Equation: Computational Modeling of Romantic Relationships in French Classical Drama

Folgert Karsdorp¹, Mike Kestemont², Christof Schöch³, and Antal van den Bosch⁴

- 1 Meertens Institute
Amsterdam, The Netherlands
fbkarsdorp@fastmail.nl
- 2 University of Antwerp
Antwerp, Belgium
mike.kestemont@uantwerp.be
- 3 University of Würzburg
Würzburg, Germany
c.schoech@gmail.com
- 4 Radboud University
Nijmegen, The Netherlands
a.vandenbosch@let.ru.nl

Abstract

We report on building a computational model of romantic relationships in a corpus of historical literary texts. We frame this task as a ranking problem in which, for a given character, we try to assign the highest rank to the character with whom (s)he is most likely to be romantically involved. As data we use a publicly available corpus of French 17th and 18th century plays (<http://www.theatre-classique.fr/>) which is well suited for this type of analysis because of the rich markup it provides (e.g. indications of characters speaking). We focus on distributional, so-called second-order features, which capture how speakers are contextually embedded in the texts. At a mean reciprocal rate (MRR) of 0.9 and MRR@1 of 0.81, our results are encouraging, suggesting that this approach might be successfully extended to other forms of social interactions in literature, such as antagonism or social power relations.

1998 ACM Subject Classification I.2.7 Natural Language Processing

Keywords and phrases French drama, social relations, neural network, representation learning

Digital Object Identifier 10.4230/OASICS.CMN.2015.98

1 Introduction

Scholarship on literary texts has been among the seminal humanistic disciplines to engage with computational approaches [17], with e.g. Burrows's well-known study of Jane Austen's novels [6]. Burrows – and many others after him – have drawn attention to the potential of computational text analysis as a viable methodological complement to established, ‘manual’ approaches in literary criticism and narratological analysis. The social relations between Austen's characters, for instance, appeared to be reflected in their language use. In general, this kind of research has raised the question of the extent to which literary concepts can be formally modeled. In this paper, we focus on the linguistic aspects of romantic relationships in literary texts. We explore how this particular kind of social relationship can be modeled. We frame this research question as a ‘matchmaking task’: given a speaker, we try to assign



© Folgert Karsdorp, Mike Kestemont, Christof Schöch, and Antal van den Bosch;
licensed under Creative Commons License CC-BY

6th Workshop on Computational Models of Narrative (CMN'15).

Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 98–107

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the highest rank to the speaker with whom (s)he is most likely to be romantically involved on the basis of linguistic features.

The relationship between fictional characters in literary works can be viewed as a social network, the computational analysis of which has been steadily gaining popularity in recent years [15, 22]. When applied to literary fiction such as novels or plays, network analysis can yield insight into character relations in individual literary works or, more interestingly, reveal patterns and structure with regard to character networks in large collections of works. In this study, we analyze a collection of French plays from the 17th and 18th centuries. Relations between speakers are a central concern in research about dramatic works (see e.g. [19]), and love relationships are a type of speaker relation present in virtually any play from the period studied here. A basic assumption underlying our research is that love relationships in fiction are not only a matter of psychology, but are also a textual phenomenon which can be derived from the language used by speakers in a play. As a consequence, this study focuses on developing new methods for the formal modeling of love relationships in dramatic works based on speakers' linguistic behavior.

Among earlier work in this field is Moretti's essay 'Network Theory, Plot analysis' [14], in which the author draws on network theory to discuss the network of characters in Shakespeare's *Hamlet*, reminiscent of Knuth's classic network dataset [11] representing co-appearance patterns of characters in Victor Hugo's *Les Misérables*. A series of publications in the field of computational linguistics have further advanced a similar line of research in recent years, including social network analyses of e.g. nineteenth-century fiction [9]; *Alice in Wonderland* [1, 2], topic-model based approaches [7] and authorship attribution based on network features of novels [4]. A popularizing analysis of Marvel graphic novels has been presented in [3]. Few studies have explicitly focused on the formal modeling of love relationships in literary texts. Nevertheless, a number of inspiring studies have studied other sorts of specific social interactions e.g. friend-or-foe relationships [20] or antagonism ('good guy' vs 'bad guy') often in combination with methodologies from distributional semantics [5, 16].

This paper is structured as follows. We begin with a description of the French plays we used in Section 2. We then proceed with the methodology in Section 3 in which we discuss the task description, our evaluation method, the computational system and the features we used. Section 4 discusses the results of our study after which in Section 5 we conclude with some final remarks and starting points for further research.

2 The Data

The data for this study comes from the *Théâtre classique* collection of French drama [10]. The collection contains 720 plays first published between 1610 and 1802, amounting to around 9.3 million word tokens. The plays vary in genre (with 340 comedies, 189 tragedies and 191 other sub-genres) and form (with 441 plays written in verse and 209 in prose only). The vast majority of plays have either one or five acts and 20–35 scenes. The plays are available as highly structured XML data encoded according to the guidelines of the Text Encoding Initiative (TEI P5) [8].¹ Each play's structure, in terms of acts and scenes, the cast members (henceforth, speakers) present in each scene, and their speeches, has been encoded in this markup. In addition, the XML files include detailed metadata about many of the roughly 6,500 speakers in the plays. In particular, the speakers' gender as well as their status with

¹ <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/DR.html>

regard to love relationships have in many cases been explicitly encoded in the cast list, or can be inferred from the description of speakers in the cast list, as in the following example from Molière's *Le Dépit Amoureux*:

```
<castList>
  <castItem><role id="ERASTE" civil='M' type="H" statut='aristocrate' age='A'
    stat_amour='amoureux'>\{E}RASTE</role>, amant de Lucile.</castItem>
  <castItem><role id="LUCILE" civil='F' type="H" statut='aristocrate' age='A'
    stat_amour='n'\{e}ant'>LUCILE</role>, fille d'Albert.</castItem>
[...]
```

For the analyses presented here, we only used plays in which either such explicit annotation is available, or where it was possible to extract such information from the text provided in the cast list. Depending on the information available, we marked love relationships as either reciprocal or unidirectional. We extracted 295 love relationships from 200 different plays, of which only 90 could be assumed to be reciprocal. We created two datasets: one containing the 90 reciprocal relations, and one containing all 295 relationships, including all cases of unrequited love. We report results on both datasets.

3 Methods

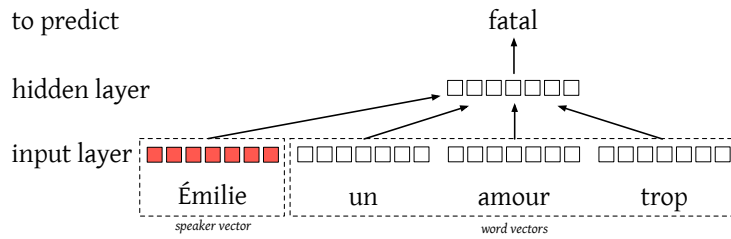
Task Description. We cast our matchmaking problem as a ranking problem. Given a query speaker s_q from a particular play, the system should return a ranking of all other speakers in that play. The goal is to produce a ranking in which the highest rank is allocated to the true lover s_j . Framing our task as a ranking problem allows us to inspect the relation between a target speaker and the second-ranked speaker, who may be a contestant of the first-ranked speaker.

Learning to Rank. Learning to Rank is a supervised machine learning task which is to learn a ranking from observed data. Learning to Rank offers a simple, yet effective way to include heterogeneous features in one model. We make use of the *sofia-ml* toolkit [18] with the *pegasos* learning algorithm and the regularization parameter at its default value ($\lambda = 0.1$). As the algorithm randomly presents samples to the ranker, each run could produce slightly different results. All scores reported in this study are obtained by running the algorithm ten times with different random seeds, and taking the average over the results.

Evaluation. We test the performance of our system by means of leave-one-lover-out cross-validation. The training and test data are constructed in such a way that the query speaker s_q is only present in the test data and no relations to s_q are included in the training data. We evaluate our approach by means of the evaluation metric Mean Reciprocal Rank (MRR) [21] which computes the reciprocal of the rank at which the first relevant speaker (the true lover) was retrieved. MRR is a natural choice for our problem since in general, each speaker is at most in love with one other person. To evaluate the accuracy of the model we compute the MRR with a rank cutoff at 1.

3.1 Features

For each speaker in a play, we extract a vector containing the features described below. We scale each feature x within each query to the range $0 \leq x \leq 1$.



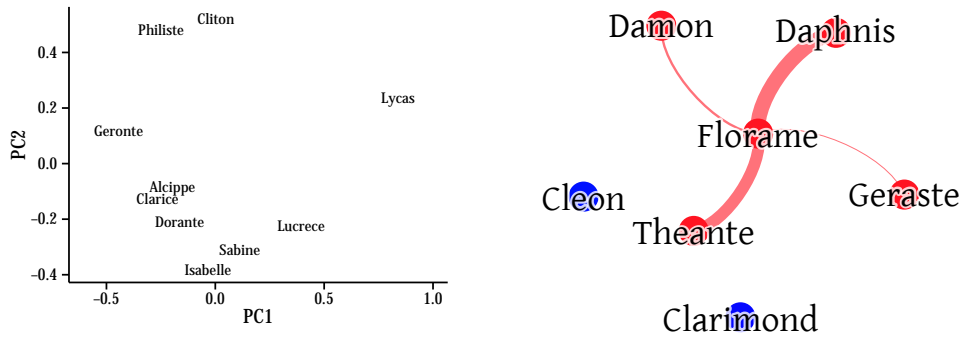
■ **Figure 1** Graphical illustration of the Speaker Vector Model. In the model, each speaker has a unique vector representation (here milie in red). The speaker vector is activated each time a speaker generates an utterance.

3.1.1 Speaker Vectors

The first two features aim to capture information about the relationship between two speakers on the basis of their distributional semantics. For each speaker we want to learn a representation that aims to capture their semantic behavioral properties, such as the topics they speak of or the people they speak or think of. The approach we take to learn such representations is inspired by the recently proposed *Paragraph Vector* model [12]. This model is a shallow neural network that aims to learn dense, fixed-length semantic representations for arbitrarily long pieces of text. In the model, each paragraph (or any other chosen text unit, e.g. sentences or complete documents) is mapped to a unique vector of n dimensions. The words in the paragraphs are also mapped to a vector. However, these vectors are shared across word tokens, hence are not unique. The model initializes all vectors randomly. It then attempts to update the values along the dimensions by continuously predicting the next word in a particular context on the basis of these vectors. All vectors are trained using stochastic gradient descent. The dimensions (parameters) are updated by back-propagating the gradient through the network.

Our model learns dense representations not for individual paragraphs but for speakers. It does so in much the same way as the *Paragraph Vector* model, the only difference being that whereas the paragraphs in the original model are represented by a unique vector, a paragraph in our *Speaker Vector* model is mapped to the vector that belongs to the speaker of that paragraph. Figure 1 provides a graphical illustration of the model. The vector in red represents the vector of the speaker *Émilie*. Together with the context vectors for *un*, *amour* and *trop* the model attempts to predict the word *fatal*. The speaker vector of a speaker is activated during each utterance of that speaker and is used to predict each word in that utterance.

F1. Speaker Similarity For each candidate lover $s \in S$, where S is the set of candidate lovers in a play, we compute the cosine similarity between its vector representation and the vector representation of a query speaker s_q , $s_q \notin S$. The idea behind this feature is that we expect two lovers to speak of similar topics in similar ways, which should be reflected in their vector representations. To illustrate this point, in Figure 2a we present a two-dimensional reproduction of the speaker vectors in Pierre Corneille’s comedy *Le Menteur* from 1644. The dimension reduction was generated through principal component analysis (PCA). The two lovers Alcippe and Clarice are placed adjacent to each other, reflecting the similarity of their vector representations. Interestingly, Alcippe’s main contestant Dorante, the liar of the play’s title, is close by. With some imagination, the plot visually expresses their contest around their object of desire, Clarice. To investigate



(a) 2-dimensional PCA projection of the speaker vectors in Pierre Corneille's *Le Menteur*. The two lovers Alcippe and Clarice are placed adjacent to each other, reflecting the similarity of their vector representations.

(b) Interaction network of Florame in Pierre Corneille's *La Suivante*. Florame predominantly interacts with two other speakers (depicted by the edge weight) of which Daphnis is his true love. Interestingly, Florame also often interacts with Theante who also is in love with Daphnis.

■ **Figure 2** Visualization of features F1. and F5.

the overall effect of being a couple on the similarity between two speakers, we computed the pairwise cosine similarity between all lover and non-lover pairs within the same play. According to a two-sample Kolmogorov-Smirnov (KS) test, the two cosine similarity distributions differ significantly ($p < 0.0005$).

F2. Analogous Lovers The relation between Clarice and Alcippe can be described by their displacement vector D : $D(\text{Clarice}, \text{Alcippe}) = s_{\text{Clarice}} - s_{\text{Alcippe}}$, where s_{Clarice} is the vector representation of Clarice and Alcippe is represented by s_{Alcippe} . We can use this relation as a reference point to other possible relations between speakers. The similarity between a pair of displacement vectors, each describing a particular relation, should reflect the similarity between these relations. Given the relation between e.g. Clarice and Alcippe, we can compare other relations between speakers to this relation. Relations that are similar to that of Clarice and Alcippe are assumed to be romantic relationships. An illustrative example is the relation between Rosidor and Caliste from Pierre Corneille's highly complex early tragi-comedy *Clitandre*, first performed in 1630. Of all relations between Rosidor and any other speaker in the play, the one with Caliste is the one that is most similar to the relation between Clarice and Alcippe. We use this information in the following way. For each candidate lover $s \in S$ and a query speaker s_q , we compute the cosine similarity between the displacement vector $D(s, s_q)$ and the displacement vectors of all known lover couples. The maximum similarity between $D(s, s_q)$ and any other pair is used as the feature value. To assess the overall similarity between couples versus non-couples, we computed the maximum similarity between the displacement vectors of lover pairs to all other lover pairs and all non-lovers to all lover pairs. Again, the similarity distributions are significantly different (KS: $p < 0.0005$).

3.1.2 Word Vectors

Speaker vectors aim to capture topical properties of speakers. The similarity between two speaker vectors reflects the extent to which the two speakers speak of similar topics. Lovers also tend to speak about each other and often third parties talk about a couple. Speaker vectors do not necessarily capture this information, because most text in plays is in direct

speech in which speakers refer to themselves by means of pronouns. To model the textual proximity of speakers we construct a version of the corpus in which each first person pronoun (*je, me, moi, mon, ma*) has been replaced by the unique ID of the speaker it refers to. Because speakers with the same name act in different plays, we also replace all proper names with the same unique ID. Essentially, this procedure is a cheap method to resolve co-references. We train word vectors on these adapted texts with 200 dimensions using the skip-gram and CBOW architecture [13].

F3. Word Similarity Similar to F1., for each candidate lover $s \in S$ we compute the cosine similarity between his/her word vector representation and the word vector representation of a query speaker s_q , $s_q \notin S$. On average, lovers have a cosine similarity of 0.58 while the mean cosine similarity between non-lovers is 0.34. As with the previous features, the similarity distributions are significantly different (KS: $p < 0.0005$).

F4. Word Analogy In a similar way as F2., we compute the maximum cosine similarity between the displacement vector $D(s, s_q)$ for candidate lover s and query speaker s_q and the displacement vectors of all known love couples. (KS: $p < 0.005$)

3.1.3 Physical Co-occurrence Features

The speaker vectors capture topical similarities and co-occurrence features present in the text. Not necessarily do these features reflect the physical co-occurrence of two speakers, for instance in a particular scene. The following two features aim to capture the physical co-occurrence of speakers. The idea behind these features is that two speakers are more likely to be in a love relationship if they meet more often.

F5. Interaction Frequency The first physical co-occurrence feature estimates the frequency of interaction between two speakers. Speaker s_i is in interaction with s_j if an utterance of s_i is preceded or followed by an utterance of s_j . For each speaker we compute the normalized count of how often (s)he interacts with another speaker. The result can be described as a network for each speaker in which weighted edges between two speakers are created if they interact. Edge weights are determined by the frequency with which the speakers interact. Figure 2b provides a graphical illustration of this feature in which we show the interaction network of Florame from Pierre Corneille's five-act comedy *La Suivante*, first performed in 1634. Florame predominantly interacts with two other speakers (depicted by the edge thickness) of which Daphnis is his lover. Interestingly, Florame also often interacts with Theante who is also in love with Daphnis. The overall interaction frequency distribution differences between couples and non-couples is significant (KS: $p < 0.0001$).

F6. Scene Co-occurrence The second physical co-occurrence feature is similar to F5. Here we construct a co-occurrence network for each speaker in a play in which edges between speakers are created if they appear in the same scene. The distribution differences between couples and non-couples are again significant (KS: $p < 0.0001$).

3.1.4 Meta Features

The XML-formatted versions of our plays provide rich metadata. One of the annotated features is the gender for each speaker. Given the dominance of heterosexual relationships in 17th and 18th century plays, we can apply an *a priori* filter on possible lover candidates on the basis of gender. To allow our system to be employed for different corpora that show

■ **Table 1** Feature performance investigation. The first four columns provide the performance of the system with (individual) features on the full data set and the reciprocal data set. The last four columns show the performance of the system after removing the features mentioned.

feature	with feature				without feature			
	Reciprocal		Full		Reciprocal		Full	
	MRR	@1	MRR	@1	MRR	@1	MRR	@1
F1. Speaker Similarity	0.51	0.29	0.51	0.28	0.89	0.79	0.86	0.74
F2. Analogous Lovers	0.41	0.18	0.48	0.27	0.87	0.76	0.86	0.74
F3. Word Similarity	0.74	0.59	0.73	0.56	0.77	0.60	0.79	0.64
F4. Word Analogy	0.45	0.24	0.41	0.22	0.88	0.77	0.86	0.74
F5. Interaction Frequency	0.53	0.28	0.55	0.32	0.88	0.78	0.87	0.77
F6. Scene Co-occurrence	0.53	0.32	0.51	0.28	0.87	0.74	0.87	0.75
F7. Gender	0.29	0.07	0.37	0.12	0.71	0.50	0.71	0.52
F1. – F7.	0.9	0.81	0.87	0.75	–			

more variability in terms of the nature of relationships, we encode the gender of speakers as a feature.

F7. Gender For each combination of candidate lover $s \in S$ and the query speaker s_q , we compare their gender, where a gender difference is represented by a value 1 and gender identity by 0.

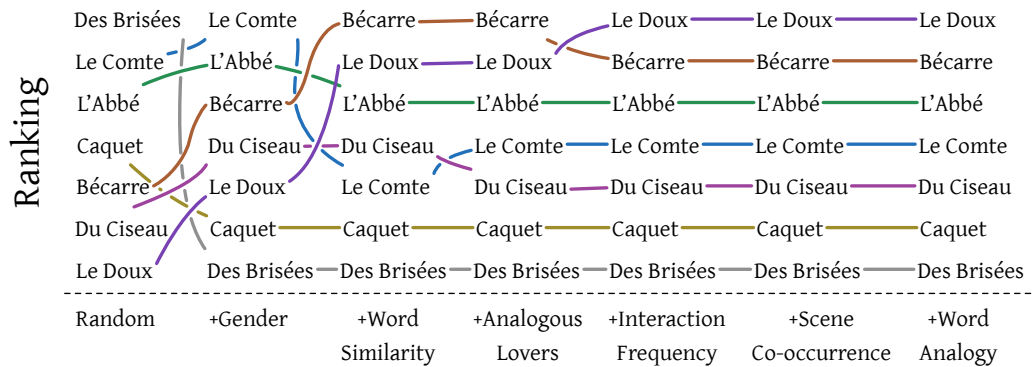
4 Results

Our Learning to Rank system shows promising results. The system achieves a Mean Reciprocal Rank of 0.9 on the dataset containing solely reciprocal love relationships and 0.87 on the full dataset. The MRR@1 (or accuracy) of the model on the reciprocal relationships is 0.81 and 0.75 on the full data set.

We performed an additional experiment in which for each feature we train our system using only that feature. The features in a Learning to Rank system can interact with each other in non-linear ways, implying that features that appear to have little effect in isolation may contribute strongly to the overall performance in combination with other features. We therefore also performed an ablation experiment in which for each feature we trained a system on the basis of all features except that feature. In Table 1 we present the results of the experiment that measures the performance of individual features (first four columns) and the results for the ablation experiment (last four columns).

In both the full data set and the data set containing solely reciprocal love relationships, the Word Similarity feature (F3.) is the best individually performing feature. The physical co-occurrence features (F4. and F5.) come next, followed by the Speaker Similarity feature (F1.) and the analogy-based features (F2. and F4.) The low performance of the gender feature is no surprise because it selects a number of speakers yet is unable to discriminate between them. In contrast, in the ablation experiment gender has the biggest contribution to the performance. Without the gender feature, the MRR drops from 0.9 to 0.71.²

² Note that this score is even lower than the score obtained by the Word Similarity alone. This suggests



■ **Figure 3** The effect of cumulatively adding features. We start with a random ranking (no features) and then continuously add more features. The plot visualizes how Le Doux, the lover of Suzanne in Beaunoir’s *Le Sculpteur* gradually rises to the first position.

The gender feature acts as a sort of funnel that makes a pre-selection among possible love candidates. Given this pre-selection, the system makes a decision on the basis of the other features. To illustrate this process, we provide in Figure 3 the different rankings produced by the system for one speaker, Suzanne from Madame de Beaunoir’s two-act prose comedy *Le Sculpteur* first performed in 1784. We start with a random ranking. The next ranking is based solely on the gender feature and puts all male speakers in the highest positions. As we add more features, Suzanne’s lover Le Doux slowly rises to higher positions and takes over the first position from Bécarré when we add feature F5. Interaction Frequency.

5 Conclusions

The system for identifying romantic relationships in drama texts introduced here proves to be successful. We have shown that on the basis of textual and structural distributional properties of speakers in French drama texts we are able to confidently extract love relationships between speakers from the texts. These distributional properties function best in combination with knowledge about the gender of two speakers. Since knowledge about the gender of a potential couple is so important to our model and because we rely on manual annotations of this feature, the first point of future research should be the automatic classification of speaker gender. Next, we believe that our approach might be a fruitful starting point for modeling other relationships, such as well-known relations from structuralist analyses of drama, such as the triangle of protagonist, helper and antagonist [19].

One important limitation of the present setup is that the system can naively assume that all analyzed speakers are at least involved in one romantic relationship. The task is thus to identify, for a given speaker, the correct lover among a set of candidates. A more general, yet also more demanding task would be to predict for any given character, whether (s)he is romantically involved at all with another character. The distinction between both tasks is reminiscent of the difference between authorship attribution and authorship verification. With the former, resembling a police line-up, the system can assume that the correct author is present among the candidates. In the verification setup, however, the correct author is

that there are some interactions between features that actually harm the overall performance. We plan to investigate this in future work.

not necessarily included among the candidates. In future research, we hope to be able to generalize our model in this respect.

Our method could more generally serve as a heuristic tool for the exploration of large literary corpora and the serendipitous discovery of unsuspected speaker relations. Its ranking fosters investigations, for example, into what types of relations there are between the target speaker and the second-ranked speaker, who may for instance be a rival or a family member of the first-ranked speaker. More generally, our method is relevant in the context of increasing amounts of literary texts becoming available through large-scale digitization of our cultural heritage. Such textual data does not usually contain the rich annotations our data contains, and manually adding it is labor-intensive. Automatically extracting fundamental speaker relationships from raw text versions of plays helps gain a hermeneutically valuable access to such ever larger amounts of textual data.

Acknowledgments. The work of Folger Karsdorp and Antal van den Bosch has been supported by the Computational Humanities Programme of the Royal Netherlands Academy of Arts and Sciences, under the auspices of the Tunes & Tales project. For further information, see <http://ehumanities.nl>. Mike Kestemont has been supported for this work as a postdoctoral researcher for the Research Foundation Flanders (FWO). Christof Schöch's contribution has been supported by funding from the German Federal Ministry of Education and Research under the eHumanities scheme (funding code 01UG1408); for more information, see <http://clgs.hypotheses.org/>

References

- 1 Apoorv Agarwal, Augusto Corvalan, Jacob Jensen, and Owen Rambow. Social network analysis of alice in wonderland. In *The proceedings of Workshop on Computational Linguistics for Literature, NAACL 2012*, pages 88–96, Montréal, Canada, 2012.
- 2 Apoorv Agarwal, Anup Kotalwar, and Owen Rambow. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013)*, pages 1202—1208, Nagoya, Japan, 2013.
- 3 Ricardo Alberich, Joe Miro-Julia, and Francesc Rosselló. Marvel universe looks almost like a real social network. Preprint, arXiv id 0202174, 2002.
- 4 Mariona Coll Ardanuy and Caroline Sporleder. Structure-based clustering of novels. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 31–39, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.
- 5 David Bamman, Brendan O'Connor, and Noah Smith. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 352—361, Sofia, Bulgaria, 2013.
- 6 John Burrows. *Computation into criticism. A Study of Jane Austen's novels and an experiment in methods*. Clarendon Press, 1987.
- 7 Asli Celikyilmaz, Dilek Hakkani-Tur, Hua He, Greg Kondrak, and Denilson Barbosa. The actor-topic model for extracting social networks in literary narrative. In *NIPS Workshop: Machine Learning for Social Computing*, 2010.
- 8 TEI Consortium. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. TEI Consortium, 2014.
- 9 David K. Elson, Nicholas Dames, and Kathleen R. McKeown. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden, 2010.
- 10 Paul Fievre, editor. *Théâtre classique*. Université Paris-IV Sorbonne, 2007-2014.

- 11 Donald Ervin Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. ACM Press Series. ACM Press, 1993.
- 12 Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 2014.
- 13 Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.
- 14 Franco Moretti. Network theory, plot analysis. *New Left Review*, 68:80–102, 2011.
- 15 Mark Newman. *Networks. An Introduction*. Oxford University Press, New York, NY, USA, 2010.
- 16 Gabriel Recchia, Alexandra L. Slater, and Max M. Louwerse. Predicting the good guy and the bad guy: Attitudes are encoded in language statistics. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pages 1264–1269, 2014.
- 17 Susan Schreibman and Ray Siemens, editors. *A Companion to Digital Literary Studies*. Oxford: Blackwell, 2008.
- 18 D. Sculley. Large scale learning to rank. In *NIPS Workshop on Advances in Ranking*, pages 58–63, 2009.
- 19 Anne Ubersfeld, Frank Collins, Paul Perron, and Patrick Debbèche. *Reading Theatre*. Toronto Studies in Semiotics and Communication Series. University of Toronto Press, 1999.
- 20 Matje Van de Camp and Antal Van den Bosch. The socialist network. *Decision Support Systems*, 53(4):761–769, 2012.
- 21 Ellen M Voorhees and Dawn M Tice. The TREC-8 question answering track evaluation. In *Proceedings of the Eighth Text REtrieval Conference (TREC 8)*, volume 1999, page 82, 1999.
- 22 Scott Weingart. Demystifying networks, parts I & II. *Journal of Digital Humanities*, 1(1):9–21, 2012.